D. REMARKS

Status of the Claims

Claims 1-32 are currently present in the Application, and claims 1, 10, 19, and 28-32 are independent claims. No claims have been amended, cancelled, or added in this Response.

Drawings

Applicants note with appreciation the acceptance, by the Examiner, of Applicants' formal drawings that were filed with the instant application.

Claim Objections Under 35 U.S.C. § 112

Claims 1, 3, 10, 12, 19, 21, and 28-32 stand rejected under 35 U.S.C. § 112, first paragraph, as failing to comply with the written description requirement. In particular, the Office Action rejects these claims due to the recitation of the claim element "extension file." Applicants respectfully traverse the rejection.

Throughout the specification, Applicants teach use of an extension file called a "plug-in." Plug-in files are well known in the art (albeit, not the plug-in files that perform the specific functions described by Applicants). Further, those skilled in the art readily understand that a plug-in file is a type of extension file.

To show that the term "plug-in" is readily understood to be a type of extension file, Applicants have provided a definition of "plug-in" from the web site titled "Free On Line Dictionary of Computing" (FOLDOC). The url for this website is: http://wombat.doc.ic.ac.uk/foldoc/contents.html.

plug-in

> <tool> A file containing data used to alter, enhance, or *extend* the operation of a parent application program. One of the first uses of this term was in Silicon Beach's SuperPaint application (late 1980s?) for the Macintosh. It had a Plug-ins folder containing different tools and effects.
>
> The Netscape Navigator World-Wide Web browser supports plug-ins which display or interpret a particular file format or protocol such as Shockwave, RealAudio, Adobe Systems, Inc. PDF,

> Corel CMX (vector graphics). The file to be displayed is included in a web page using an EMBED HTML tag.
>
> Plug-ins, both commercially and independently authored, can usually be downloaded for free and are stored locally. Plug-ins come in different versions specific to particular operating systems (Microsoft Windows 3.1, 3.2, and Macintosh are available).
>
> (emphasis added)

In other words, it was well known at the time of Applicants' invention that a "plug-in" file is a type of file that "extends" the operation of a parent application. In addition, Applicants respectfully remind the Examiner that Applicants' originally filed claims are part of Applicants' specification (*see* 37 CFR § 1.75 and MPEP § 608.01(l)). In particular, MPEP § 608.01(l) provides:

> **608.01(l) Original Claims**
>
> In establishing a disclosure, applicant may rely not only on the description and drawing as filed but also on the original claims if their content justifies it.
>
> Where subject matter not shown in the drawing or described in the description is claimed in the application as filed, and such original claim itself constitutes a clear disclosure of this subject matter, then the claim should be treated on its merits, and requirement made to amend the drawing and description to show this subject matter. The claim should not be attacked either by objection or rejection because this subject matter is lacking in the drawing and description. ***It is the drawing and description that are defective, not the claim.***
>
> It is, of course, to be understood that this disclosure in the claim must be sufficiently specific and detailed to support the necessary amendment of the drawing and description.
>
> (emphasis added)

Applicants submit that the use of claim elements "extension file" in the independent claims (e.g., claim 1) and "plug-in file" in dependent claims (e.g., dependent claim 3) makes it perfectly clear that a "plug-in" file, used to describe Applicants' embodiment in the specification, is one type of an "extension file." Indeed, claim 3, which depends from claim 1, further defines an "extension file" as including a "plug-in file."

Applicants have amended the detailed description to describe "extension files" as including "plug-in files." No new matter has be added by this amendment because support is found in Applicants' original claims. As MPEP § 608.01(l) provides, if any defect existed in Applicants' disclosure regarding the term "extension file," such defect was in the description and not in the claims. Therefore, Applicants' claims should not have been "attacked" by objection or rejection. Instead, the Office Action should have required that Applicants' specification be amended to show this subject matter.

In light of the foregoing, and in light of Applicants' amendment, Applicants respectfully request that the Examiner withdraw the rejection of claims 1, 3, 10, 12, 19, 21, and 28-32 under 35 U.S.C. § 112, first paragraph, as moot.

Claim Rejections – Alleged Obviousness Under 35 U.S.C. § 103

Claims 1-6, 8-15, 17-24, 26-28, and 30-32 stand rejected under 35 U.S.C. § 103(a) as being obvious, and therefore unpatentable over U.S. Pat. Appl. Publ. No. 2003/0095145 to Patrizio et al. (hereinafter "Patrizio"), in view of U.S. Pat. No. 6,311,321 to Agnihotri et. al. (hereinafter "Agnihotri"). The remaining claims, 7, 16, 25, and 29, stand rejected under 35 U.S.C. § 103 as being obvious, and therefore unpatentable, over Patrizio in view of Agnihotri in further view of U.S. Pat. Appl. Publ. No. 2002/0103660 to Cramon et al. (hereinafter "Cramon"). Applicants respectfully traverse the rejections.

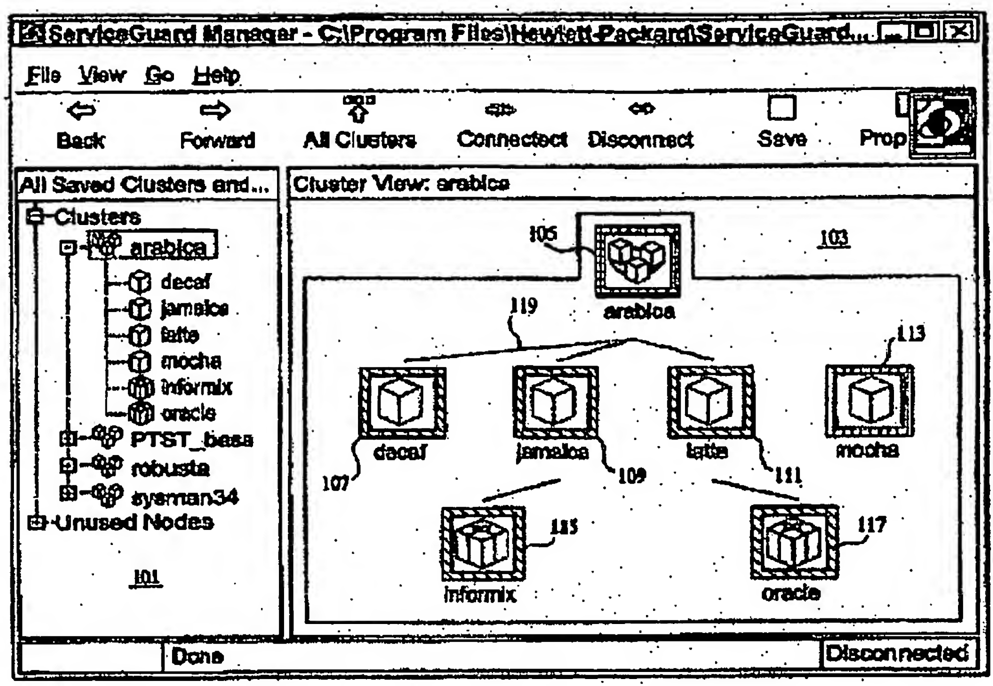
Independent claims 1, 10, and 19 claim a method/system/ program product for converting management models to one or more console interfaces. Each of these independent claims includes the limitations of:

- receiving a console selection corresponding to one of the console interfaces;
- identifying one or more console algorithms corresponding to the console selection;
- retrieving a generic management object from a management definition object; and
- creating an extension file adapted to perform the generic management object on the selected console interface.

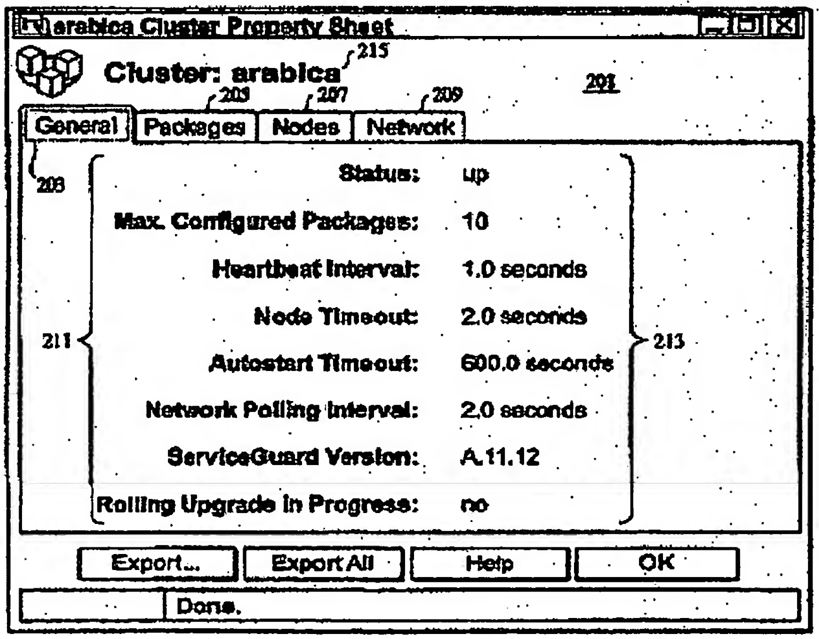
The Office Action mistakenly contends that Patrizio teaches each of these limitations. Applicants respectfully disagree.

First, regarding the limitation of *receiving a console selection corresponding to one of the console interfaces,* the Office Action does not even recite the limitation as claimed. Instead, the Office Action states that Patrizio teaches "receiving a selection window corresponding to one of the graphical user interface panels." In essence, the Office Action is rejecting a claim limitation that ***has not*** been claimed by Applicants. A "console interface," as used in Applicants' disclosure is an interface provided by a "management console" which "allow common applications from which a user views and manipulates data associated with a software program or device that is accessible by the console." (see, e.g., page 3, line 27 to page 4, line 13). Applicants simply ***are not*** claiming "receiving a selection window corresponding to one of the graphical user interface panels," as suggested by the Office Action's rejection. First, Applicants' claim is for receiving "a console selection," not "a selection window." Second, Applicants' claimed "selection" corresponds to "one of the console interfaces," not to "one of the graphical user interface panels."

The Office Action cites Figures 1 and 2 from Patrizio's reference to support this unfounded rejection. These figures are reproduced below. Figure 1 shows a top-level screen shot from a "ServiceGuard Manager" product, and Figure 2 shows a screen shot of a "General" tab from a cluster property sheet. A review of these figures reveals that neither teaches or suggests a "console selection." In fact, while Applicants' claims are directed to "console interfaces" (explained thoroughly in Applicants' specification), the term "console" ***never even appears*** in the Patrizio reference.



Patrizio's Figure 1



Cluster: arabica

Status:	up
Max. Configured Packages:	10
Heartbeat Interval:	1.0 seconds
Node Timeout:	2.0 seconds
Autostart Timeout:	600.0 seconds
Network Polling Interval:	2.0 seconds
ServiceGuard Version:	A.11.12
Rolling Upgrade In Progress:	no

Patrizio's Figure 2

Moreover, the "ServiceGuard" product discussed in Patrizio is described as being a "specialized facility for protecting mission-critical applications from a wide variety of hardware and software failures," and is not described as being a management console. In addition, Applicants reviewed the Patrizio reference and found that Patrizio does not teach or suggest "plug-ins" or ways to extend the functionality of the ServiceGuard product described by Patrizio. Extending the console using plug-in, or extension, files is a key aspect of Applicants' claimed invention and is completely missing from Patrizio.

The Office Action contends that Patrizio teaches Applicants' next limitation, "identifying one or more console algorithms corresponding to the console selection," citing various sections of Patrizio that teach structuring a managed object format (MOF) file using a unified modeling language (UML) class diagram. While Applicants' management definition object is similar to a CIM MOF file (see Applicants' claim no. 2), the point of the "identifying..." limitation is to identify console algorithms that correspond to the selected console selection. Because Patrizio does not teach or suggest "console selections," it logically follows that Patrizio cannot teach identifying algorithms that correspond to the console selection. In addition, the cited reference on page 3, para. 42, actually teaches away from Applicants' claimed limitation. The last sentence of this paragraph states: "Inside of a file ***that is internal*** to ServiceGuard Manager there is a MOF file which contains a description telling the program how a cluster property sheet should be rendered." (emphasis added). While Applicants claimed' method is for creating extension files that can perform a management object that can be used with a console interface, Patrizio appears to be teaching a system where the rendering description is "internal to" a system and, consequently, ***teaches away from*** Applicants' claimed method/system/program product that provides for the extension of the management object onto the selected console interface.

Next, the Office Action contends that Patrizio teaches or suggests Applicants' claimed limitation of "retrieving a generic management object from a management definition object," citing paragraph 41 of Patrizio. Here, Patrizio does teach using property sheets and Patrizio's property sheets are described in the MOF file. However, the shortcoming of Patrizio is not with its use of a management object from a management definition file, it is that Patrizio does not teach or suggest creating an "extension file adapted to perform the generic management object on [a] selected console interface" (Applicants' fourth limitation of claims 1, 10, and 19).

The Office Action contends that Patrizio teaches Applicants' limitation of "creating an extension file adapted to perform the generic management object on the selected console interface," citing Figure 9A of Patrizio. According to Patrizio, Figure 9A shows "a class schema defining a layout for property sheets." The Office Action notes "(Fig. 9A – Note: enlisting area wherein deriving additional sheets -- with JAVA class extended functionalities to support a GUI panel -- takes place reads on extension file). However, even if Patrizio teaches a JAVA class that supports a GUI panel "on the selected graphical user interface," this ***is not*** the limitation claimed by Applicants. Applicants' extension file is adapted to work with a ***selected*** console interface. On the other hand, the program taught by Patrizio does not teach or suggest integration with external programs using plug-ins or extension files. Instead, as described above, Patrizio teaches a system where such information is kept "internal to" the program. Moreover, while the GUI described in the property sheets can be changed, Patrizio only teaches using the property sheets to turn on/off aspects (e.g., tabs) of the GUI when the ServiceManager program is executed (see para. 45, top of page 4). Again, this teaches away from "receiving a console selection..." as there is only one management program ("ServiceManager") being taught by Patrizio. In addition, this teaches away from Applicants' claimed element of a "generic management object." Instead, Patrizio appears to teach more specific management objects that are designed and created for interaction with a specific software tool (Patrizio's "ServiceManager" program).

Evidently realizing the significant shortcomings of Patrizio, the Office Action admits that Patrizio does not teach or suggest that "the selected graphical user interface panel is selected console interface, nor does Patrizio teach selection console corresponding to console interfaces." (page 4, 5th para. of Office Action). Despite this serious admitted shortcoming of Patrizio, the Office Action attempts to resurrect the usefulness of Patrizio by contending that "the limitation of console interfaces is disclosed via Patrizio's selection of panel components to modify a certain GUI layout... As for the selection console limitation, the selection of cluster type and subdivision thereof in conjunction with properties of GUI layout to be changed suggests console selection in relation with layout characteristics." Applicants respectfully disagree with the contentions made in the Office Action. As explained in detail above, the selection of some GUI properties in a property sheet is entirely different from selection of a console interface. The GUI selection only determines graphical qualities of the display when viewed on one particular system (e.g., the SystemManager program). On the other hand, Applicants' selection of a particular console interface allows Applicants' claimed extension files to be created in a manner so that they perform management functions from the selected console interface. Nowhere does Patrizio teach or suggest that the management functions, such as the GUI panels described in Patrizio's property sheets, are created in a manner, based on a console selection, that allow the management functions to be performed on a different console. Instead, the management functions described by Patrizio are taught as being exclusively performed using Patrizio's "ServiceManager" software. In addition, Applicants' pointed out various sections of Patrizio that describe how Patrizio's property sheets are rendered using data that is "internal to" Patrizio's ServiceManager software. Patrizio does not teach or suggest using such "internal data" for use by a different program (e.g., by a management console program such as the Microsoft Management Console (MCC), the AS/400 System Console, the Tivoli Console, or other commonly used management consoles).

To overcome the admitted shortcomings of Patrizio, the Office Action contends that Agnihotri discloses the selection of a console, citing col. 5, lines 1-18 of Agnihotri. The cited section of Agnihotri discloses a "wizard application" that is used to install a product onto a selected management console. However, Agnihotri does not teach or suggest Applicants' claimed limitations of "identifying one or more console algorithms corresponding to the console selection," "retrieving a generic management object from a management definition object," or "creating an extension file adapted to perform the generic management object on the selected console interface." Indeed, the Office Action does not contend that Agnihotri teaches such limitations. What Agnihotri is describing is a well known operation performed when installing a product with a given management console. However, Agnihotri's install files and configuration files are directed to the management console selected by the user. Agnihotri does not teach or suggest using "algorithms" and "generic management objects" that are used to create "an extension file adapted to perform the generic management object on the selected console interface," as claimed by Applicants.

Applicants respectfully remind the Examiner that MPEP § 2141 sets forth, among other things, basic consideration that apply to obviousness rejections under 35 U.S.C. § 103. MPEP § 2141 states as follows:

> BASIC CONSIDERATIONS WHICH APPLY TO OBVIOUSNESS REJECTIONS
>
> When applying 35 U.S.C. 103, the following tenets of patent law must be adhered to:
>
> (A) The claimed invention ***must be*** considered as a whole;
>
> (B) The references must be considered as a whole and ***must suggest*** the desirability and thus the obviousness of making the combination;
>
> (C) The references ***must be*** viewed without the benefit of impermissible hindsight vision afforded by the claimed invention; and
>
> (D) Reasonable expectation of success is the standard with which obviousness is determined.
>
> *Hodosh v. Block Drug Co.*; Inc., 786 F.2d 1136, 1143 n.5, 229 USPQ 182, 187 n.5 (Fed. Cir. 1986)(emphasis added).

Applicants respectfully submit that Applicants' claims were not considered as a whole in the Office Action. As described above, the Office Action rejected some of Applicants' claimed limitations by improperly substituting terms found in Patrizio with Applicants' claim limitations, even though the terms and the limitations are not similar or interchangeable. In particular, the Office Action describes Applicants' first limitation as "receiving a *selection window* corresponding to one of *the graphical user interface panels*," while the actual claim limitation reads "receiving a *console selection* corresponding to one of *the console interfaces*.". As described above, Applicants' "console selection" is nothing like Patrizio's "selection window" and Applicants' "console interfaces" is nothing like Patrizio's graphical user interface panels." Applicants' console interfaces correspond to particular management consoles and are not graphical user interface panels.

Second, Applicants pointed out how Patrizio's teaching is for GUI elements that correspond to a particular software program. Patrizio does not teach or suggest a desirability to have the property sheets and GUI interfaces taught in Patrizio to work with other software programs, such as management consoles. In short, the references simply do not suggest the desirability, and thus the obviousness, of combining the references. The Office Action provides no basis as for why combining the GUI panels that work with Patrizio's ServiceManager software would be desired by Agnihotri, or vise versa. In fact, Applicants respectfully submit that such a combination would simply teach that multiple property sheets and configuration files would be included in Agnihotri's installation wizard and ***would not*** teach or suggest the use of console algorithms and generic management objects that could be used to create extension files that would be adapted for use by any of the consoles described by Agnihotri. Applicants' assertion is based on the fact that neither reference teaches "receiving a generic management object..." or "creating an extension file adapted to perform the generic management object on the selected console interface."

Finally, in light of the shortcomings of both of the cited references, Applicants' respectfully submit that the Office Action used impermissible hindsight in rejecting Applicants' claims. Applicants' contention is based on the fact that the references were selected even though neither reference suggests a desirability of combining the references. Patrizio teaches a system and method for a table row selection in a GUI display while Agnihotri teaches an in-context launch wrapper for integrating applications into management consoles. Evidently, the only reason that these references were selected was because the Examiner used Applicants' claim limitations as "guideposts" in selecting the references. Consequently, impermissible hindsight was used in the rejection of Applicants' claims.

Based on the foregoing, Applicants respectfully submit that the rejection of each of Applicants' independent claims 1, 10, and 19 over Patrizio in view of Agnihotri has been overcome. Therefore, claims 1, 10, and 19 are allowable over Patrizio in view of Agnihotri. Claims 2-9, 11-18, and 20-27 each depends, directly or indirectly, on claims 1, 10, and 19, respectively. Therefore, each of these claims is allowable over Patrizio in view of Agnihotri for at least the same reasons that the independent claims are allowable.

Finally, claims 28-32 are additional independent claims that each include the same limitations as claims 1, 10, and 19, as well as additional limitations found in various dependent claims. As each of these claims includes the allowable limitations found in independent claims 1, 10, and 19, as described above, each of these claims is also allowable over Patrizio in view of Agnihotri for at least the same reasons that claims 1, 10, and 19 are allowable.

Notwithstanding the allowability of claims 7, 16, 25, and 29 as set forth above, these claims include limitations directed at national language. The Office Action admits that neither

Patrizio or Agnihotri teaches use of national languages, however the Office Action contends that Cramon discloses national languages plug-ins. Once again, the Office Action provides no basis, found in the references themselves, for the desirability, and therefore obviousness, of combining Cramon with Patrizio and Agnihotri. Therefore, Applicants' respectfully submit that the Office Action used the limitations of claims 7, 16, 25, and 29 as guideposts in identifying the Cramon reference. Indeed, Cramon teaches a "generic transaction server" for use in e-Business transactions an never mentions or suggests the use of management consoles. Instead, the system described by Cramon operates over a network, such as the Internet. Furthermore, nowhere does Cramon teach or suggest management objects, such as a MOF file, or management definition objects. In light of the fact that there is absolutely no suggestion for combining the e-Business transaction server of Cramon with either the GUI panels that work with Patrizio's ServiceManager software or Agnihotri's installation wizard, Applicants respectfully submit that the combination of Cramon, Patrizio, and Agnihotri is improper as the Office Action used impermissible hindsight in forming the combination. Therefore, despite the allowability of claims 7, 16, 25, and 29 as including allowable limitations, these claims are also allowable because the combination used impermissible hindsight and, therefore, the rejection of these claims was improper.

Conclusion

As a result of the foregoing, it is asserted by Applicants that the remaining claims in the Application are in condition for allowance, and Applicants respectfully request an early allowance of such claims.

Applicants respectfully request that the Examiner contact the Applicants' attorney listed below if the Examiner believes that such a discussion would be helpful in resolving any remaining questions or issues related to this Application.

Respectfully submitted,

By Joseph T. Van Leeuwen

Joseph T. Van Leeuwen, Reg. No. 44,383
Van Leeuwen & Van Leeuwen
Attorneys for Applicant
Telephone: (512) 301-6738
Facsimile: (512) 301-6742